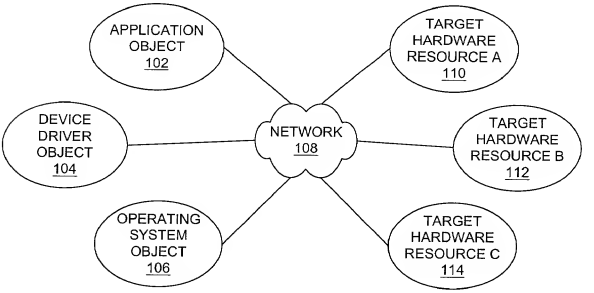

FIG. 1A
(PRIOR ART)

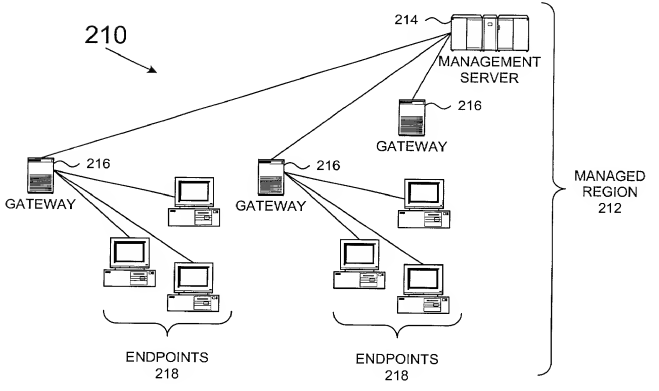

FIG. 2A

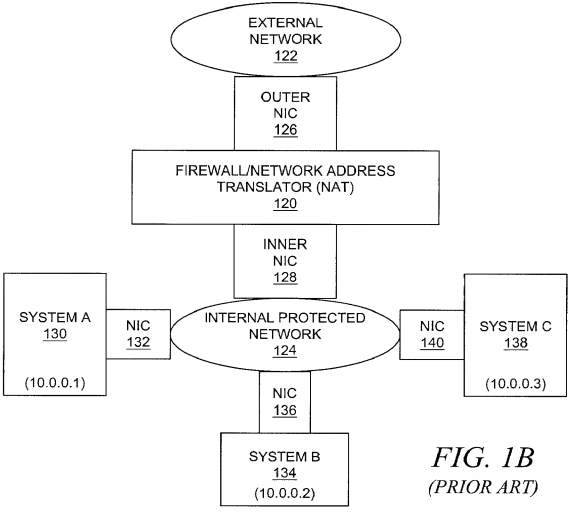

FIG. 1B
(PRIOR ART)

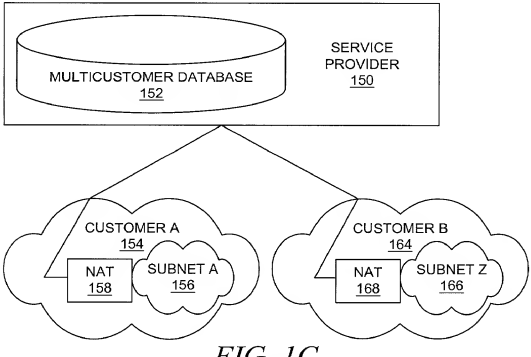

FIG. 1C

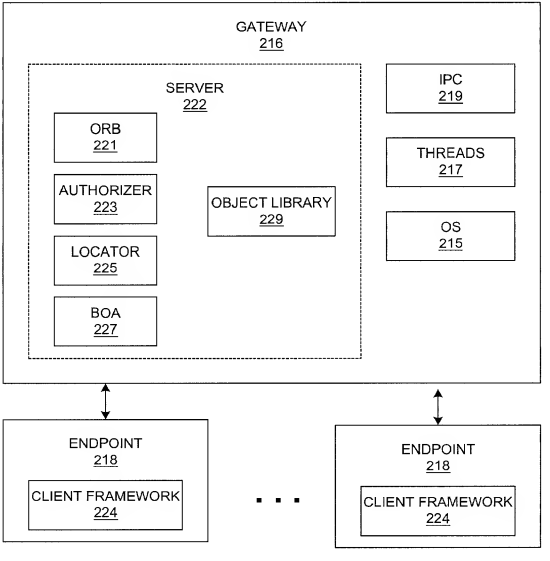
GATEWAY
216
SERVER
222
ORB
221
AUTHORIZER
223
LOCATOR
225
BOA
227
OBJECT LIBRARY
229
IPC
219
THREADS
217
OS
215
ENDPOINT
218
CLIENT FRAMEWORK
224
ENDPOINT
218
CLIENT FRAMEWORK
224

FIG. 2B

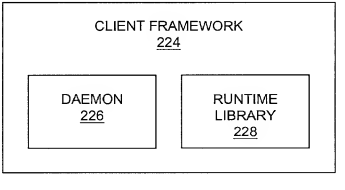
CLIENT FRAMEWORK
224
DAEMON
226
RUNTIME
LIBRARY
228

FIG. 2C

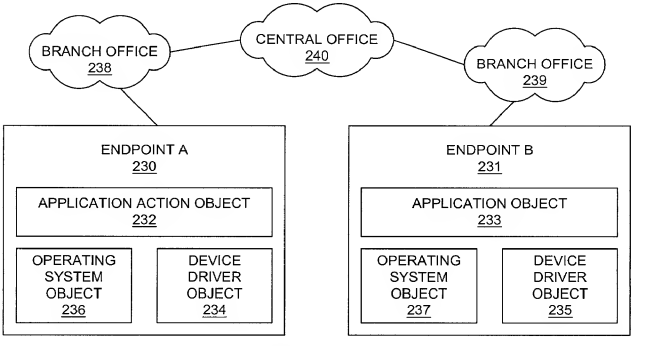
BRANCH OFFICE
238
CENTRAL OFFICE
240
BRANCH OFFICE
239
ENDPOINT A
230
APPLICATION ACTION OBJECT
232
OPERATING SYSTEM OBJECT
236
DEVICE DRIVER OBJECT
234
ENDPOINT B
231
APPLICATION OBJECT
233
OPERATING SYSTEM OBJECT
237
DEVICE DRIVER OBJECT
235

FIG. 2D

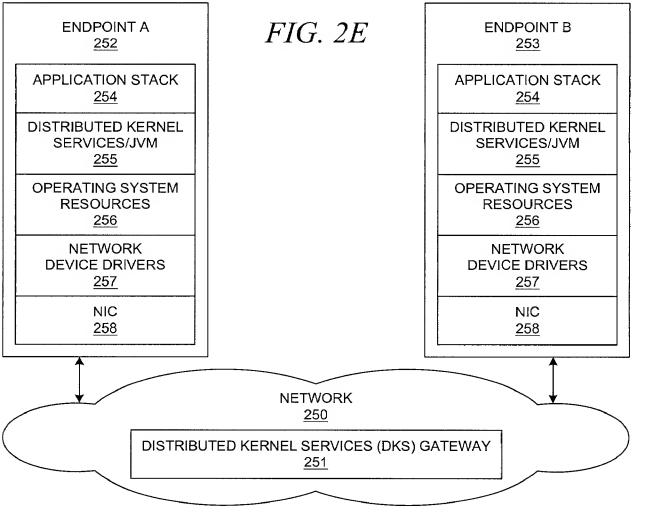
ENDPOINT A
252
APPLICATION STACK
254
DISTRIBUTED KERNEL SERVICES/JVM
255
OPERATING SYSTEM RESOURCES
256
NETWORK DEVICE DRIVERS
257
NIC
258
ENDPOINT B
253
APPLICATION STACK
254
DISTRIBUTED KERNEL SERVICES/JVM
255
OPERATING SYSTEM RESOURCES
256
NETWORK DEVICE DRIVERS
257
NIC
258
NETWORK
250
DISTRIBUTED KERNEL SERVICES (DKS) GATEWAY
251

FIG. 2E

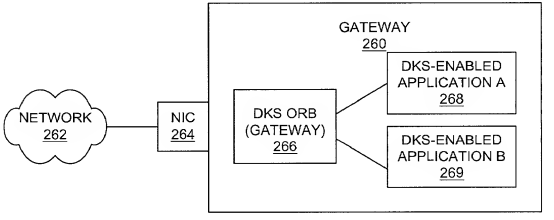
GATEWAY
260
DKS-ENABLED
APPLICATION A
268
NETWORK
262
NIC
264
DKS ORB
(GATEWAY)
266
DKS-ENABLED
APPLICATION B
269

FIG. 2F

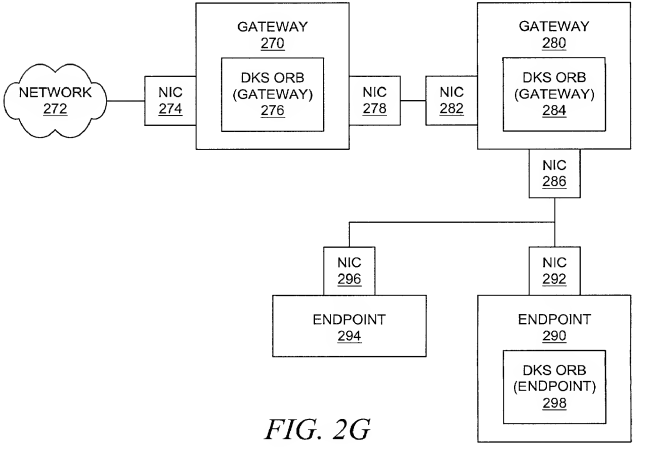
GATEWAY
270
GATEWAY
280
NETWORK
272
NIC
274
DKS ORB
(GATEWAY)
276
NIC
278
NIC
282
DKS ORB
(GATEWAY)
284
NIC
286
NIC
296
NIC
292
ENDPOINT
294
ENDPOINT
290
DKS ORB
(ENDPOINT)
298

FIG. 2G

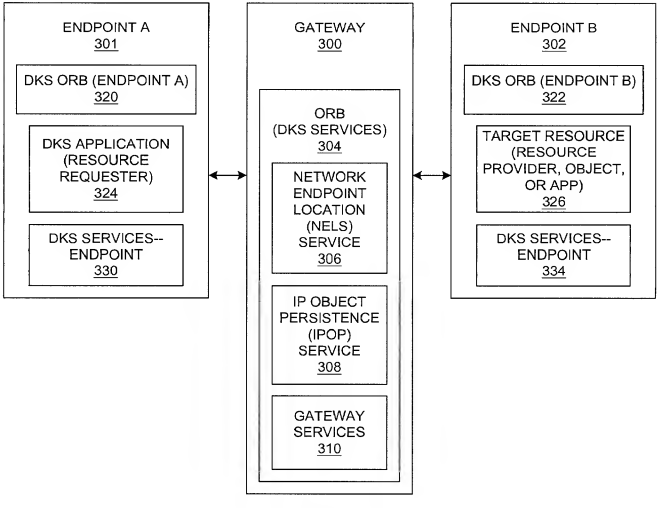
ENDPOINT A
301
DKS ORB (ENDPOINT A)
320
DKS APPLICATION (RESOURCE REQUESTER)
324
DKS SERVICES-- ENDPOINT
330
GATEWAY
300
ORB (DKS SERVICES)
304
NETWORK ENDPOINT LOCATION (NELS) SERVICE
306
IP OBJECT PERSISTENCE (IPOP) SERVICE
308
GATEWAY SERVICES
310
ENDPOINT B
302
DKS ORB (ENDPOINT B)
322
TARGET RESOURCE (RESOURCE PROVIDER, OBJECT, OR APP)
326
DKS SERVICES-- ENDPOINT
334

FIG. 3

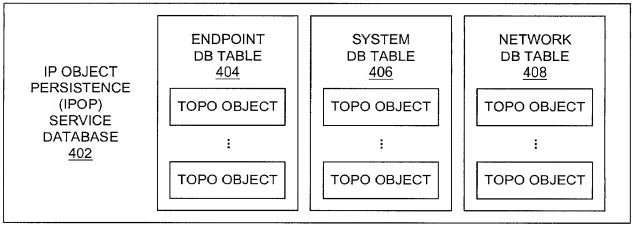
IP OBJECT PERSISTENCE (IPOP) SERVICE DATABASE
402
ENDPOINT DB TABLE
404
TOPO OBJECT
TOPO OBJECT
SYSTEM DB TABLE
406
TOPO OBJECT
TOPO OBJECT
NETWORK DB TABLE
408
TOPO OBJECT
TOPO OBJECT

FIG. 4

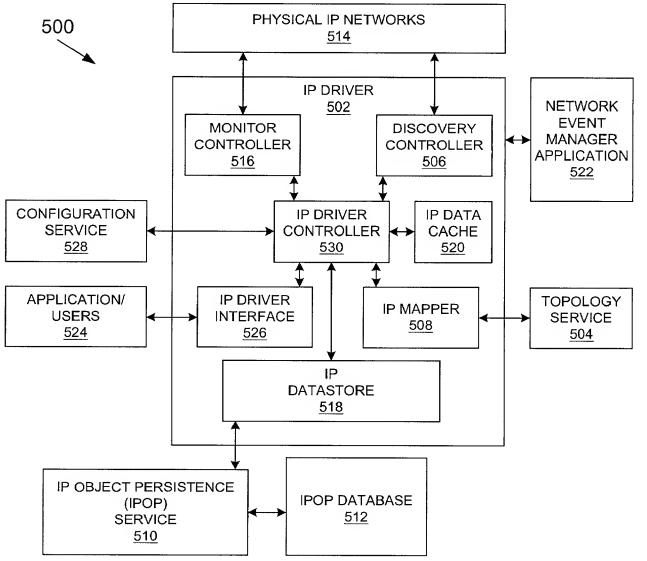
500
PHYSICAL IP NETWORKS
514
IP DRIVER
502
MONITOR CONTROLLER
516
DISCOVERY CONTROLLER
506
NETWORK EVENT MANAGER APPLICATION
522
CONFIGURATION SERVICE
528
IP DRIVER CONTROLLER
530
IP DATA CACHE
520
APPLICATION/ USERS
524
IP DRIVER INTERFACE
526
IP MAPPER
508
TOPOLOGY SERVICE
504
IP DATASTORE
518
IP OBJECT PERSISTENCE (IPOP) SERVICE
510
IPOP DATABASE
512

FIG. 5A

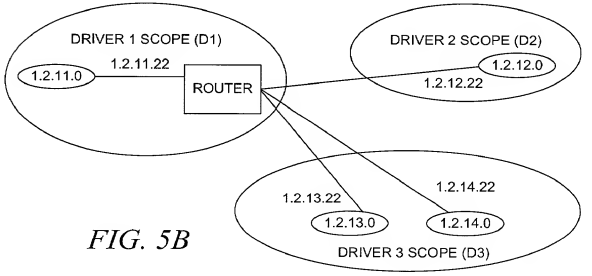
DRIVER 1 SCOPE (D1)
1.2.11.0
1.2.11.22
ROUTER
DRIVER 2 SCOPE (D2)
1.2.12.0
1.2.12.22
1.2.13.22
1.2.14.22
1.2.13.0
1.2.14.0
DRIVER 3 SCOPE (D3)

FIG. 5B

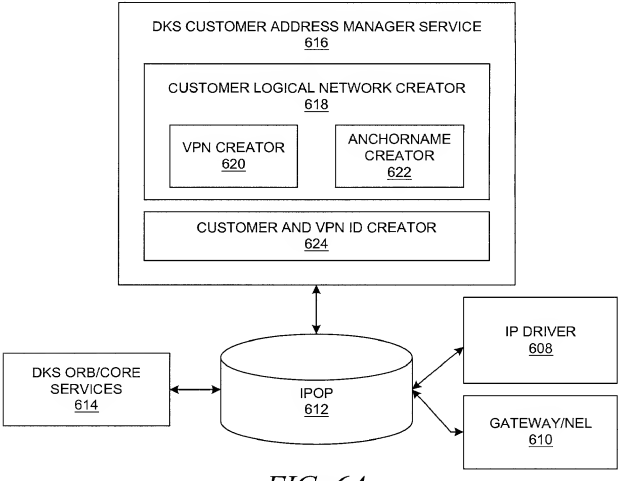
DKS CUSTOMER ADDRESS MANAGER SERVICE
616
CUSTOMER LOGICAL NETWORK CREATOR
618
VPN CREATOR
620
ANCHORNAME CREATOR
622
CUSTOMER AND VPN ID CREATOR
624
DKS ORB/CORE SERVICES
614
IPOP
612
IP DRIVER
608
GATEWAY/NEL
610

FIG. 6A

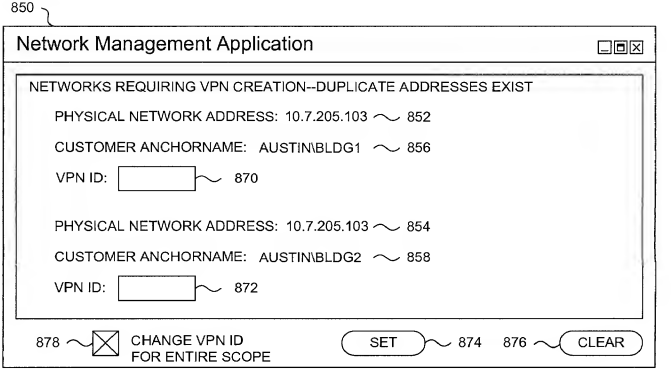
850
Network Management Application
NETWORKS REQUIRING VPN CREATION--DUPLICATE ADDRESSES EXIST
PHYSICAL NETWORK ADDRESS: 10.7.205.103
852
CUSTOMER ANCHORNAME: AUSTIN\BLDG1
856
VPN ID:
870
PHYSICAL NETWORK ADDRESS: 10.7.205.103
854
CUSTOMER ANCHORNAME: AUSTIN\BLDG2
858
VPN ID:
872
878
CHANGE VPN ID FOR ENTIRE SCOPE
SET
874
876
CLEAR

FIG. 8

```
Public Class IPActionObject {

        Endpoint sourceEP;
        Endpoint targetEP;

        // CONSTRUCTOR
        IPActionObject( Endpoint targetEP, Endpoint sourceEP ) {
        .
        .
        .
        }
        VOID performAction( ) // EXECUTES ACTION METHOD
.
.
.
}
```

FIG. 6B

```
Public Class Endpoint {

        // public variables
        long    EPObjectID; // ID to object (both private and public network addresses)
        InetAddress  EPIPAddress;       // physical network address (private or public)
        long    EPVPN;      // virtual private network ID

        //get/set of variables
        public long             getObjectID( ) { ... }
        public InetAddress      getPAddress( ) { ... }
        public long             getVPN( ) { ... }

}
```

FIG. 6C

```
Public Class EndpointCustomer extends Endpoint {

        public getVPNGW( ) {
                //gets the only gateway which has access to a particular private network
        .
        .
        .
        }
        //private variables only set/accessed by EP creator IPOP
        long    customerHashNumber;
        String  customerName;
        String  customerAnchorPath;
        Long    objectIoFPrivateGatewayRoute

}
```

FIG. 6D

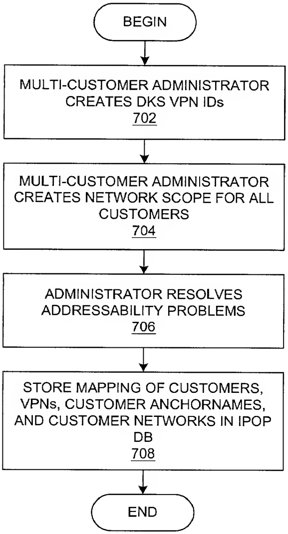
BEGIN
MULTI-CUSTOMER ADMINISTRATOR CREATES DKS VPN IDs
702
MULTI-CUSTOMER ADMINISTRATOR CREATES NETWORK SCOPE FOR ALL CUSTOMERS
704
ADMINISTRATOR RESOLVES ADDRESSABILITY PROBLEMS
706
STORE MAPPING OF CUSTOMERS, VPNs, CUSTOMER ANCHORNAMES, AND CUSTOMER NETWORKS IN IPOP DB
708
END

FIG. 7A

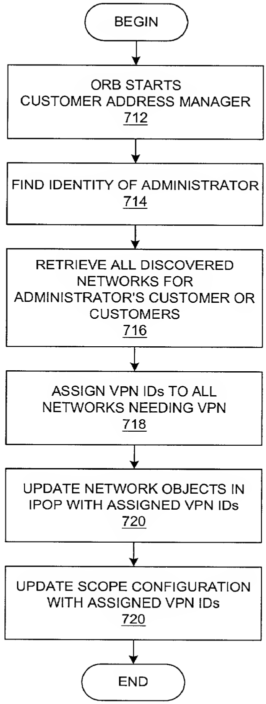
BEGIN
ORB STARTS CUSTOMER ADDRESS MANAGER
712
FIND IDENTITY OF ADMINISTRATOR
714
RETRIEVE ALL DISCOVERED NETWORKS FOR ADMINISTRATOR'S CUSTOMER OR CUSTOMERS
716
ASSIGN VPN IDs TO ALL NETWORKS NEEDING VPN
718
UPDATE NETWORK OBJECTS IN IPOP WITH ASSIGNED VPN IDs
720
UPDATE SCOPE CONFIGURATION WITH ASSIGNED VPN IDs
720
END

FIG. 7B

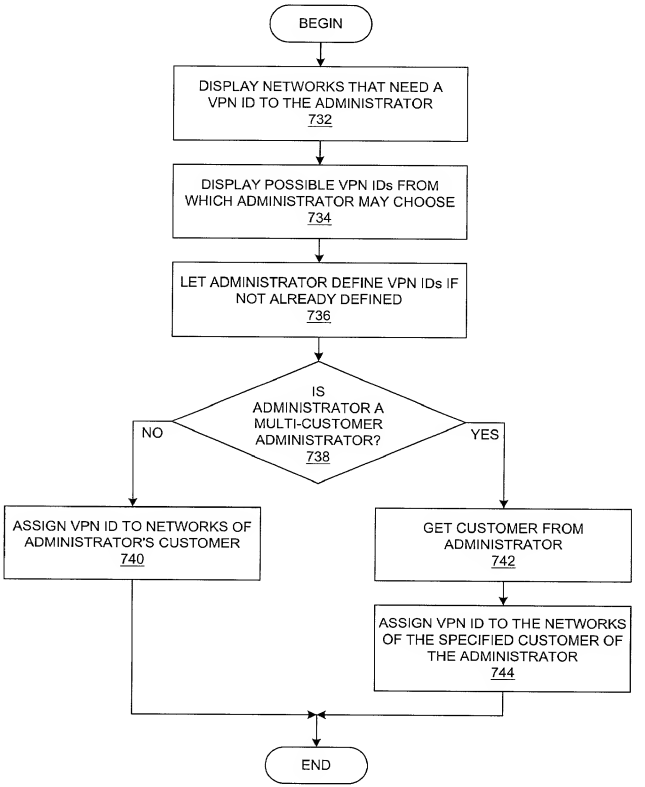
BEGIN
DISPLAY NETWORKS THAT NEED A VPN ID TO THE ADMINISTRATOR
732
DISPLAY POSSIBLE VPN IDs FROM WHICH ADMINISTRATOR MAY CHOOSE
734
LET ADMINISTRATOR DEFINE VPN IDs IF NOT ALREADY DEFINED
736
IS ADMINISTRATOR A MULTI-CUSTOMER ADMINISTRATOR?
738
NO
YES
ASSIGN VPN ID TO NETWORKS OF ADMINISTRATOR'S CUSTOMER
740
GET CUSTOMER FROM ADMINISTRATOR
742
ASSIGN VPN ID TO THE NETWORKS OF THE SPECIFIED CUSTOMER OF THE ADMINISTRATOR
744
END

FIG. 7C

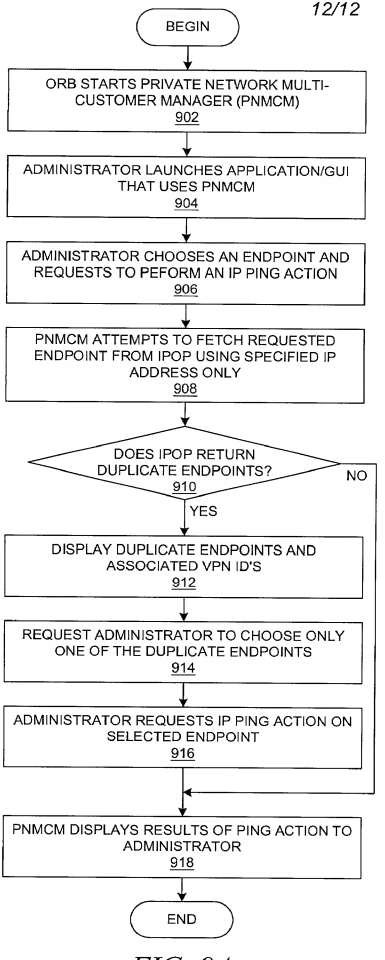

FIG. 9A

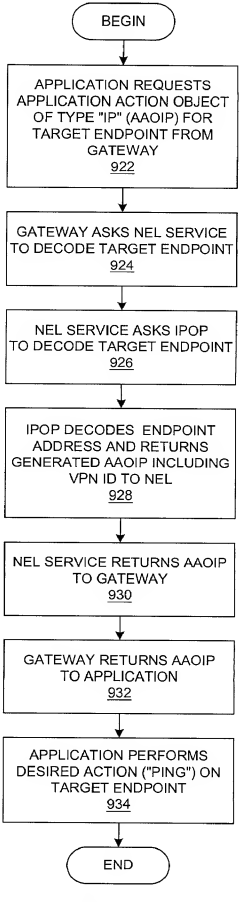

FIG. 9B